Docente: Pamela Landero Sepúlveda

ACTIVIDAD RESUELTA Arboles AVL y Heaps Binarios
Estructuras de Datos UNAB 2017-1

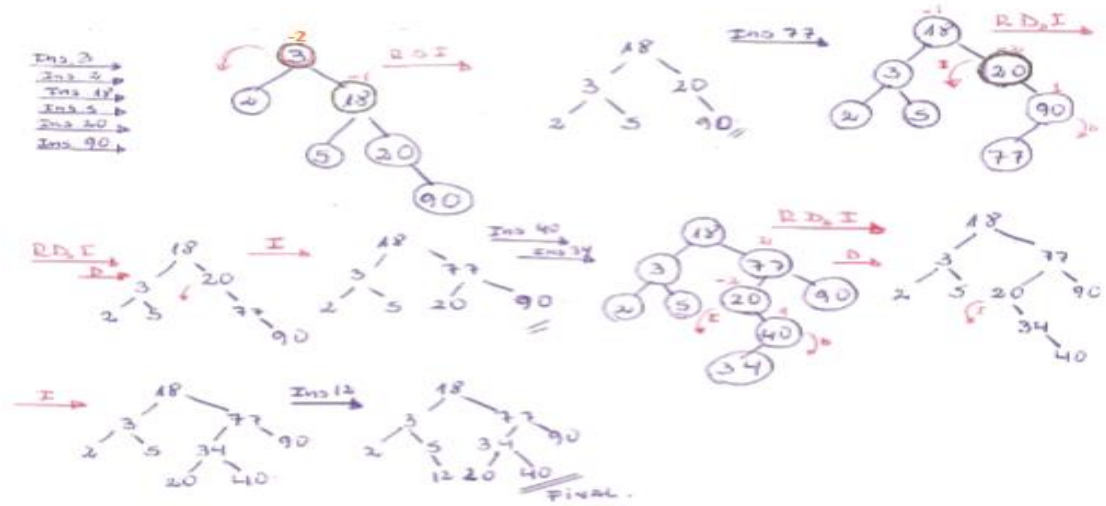
1) Cuáles son las propiedades que debe cumplir un árbol para que sea un Heap.
   - Es un árbol binario, pero **no** es de búsqueda
   - Es completo, es decir está completo hasta el último nivel y las hojas que están en el último nivel se ubican lo más a la izquierda posible)
   - Cumple criterios de ordenamiento Min-Heap (el valor del padre es menor que el de sus hijos) ó Max- Heap (el valor del padre es mayor que el de sus hijos)

(A diferencia del AVL, para eliminar se elimina el valor de la raíz reemplazándolo por el último elemento del montículo. Por otro lado, se agregan elementos, respetando la propiedad de un árbol completo. En ambos casos se respeta que cumpla la propiedad de Min-Heap o Max-Heap según corresponda).

2) Cuáles son las propiedades que debe cumplir un árbol para que sea AVL
   - Es un árbol binario de búsqueda
   - Para todo nodo del árbol la altura de sus subárboles difiere a lo más en 1 (debe estar balanceado)
   - A través del factor de equilibrio (FE) se puede saber si un árbol ha perdido su condición de AVL. Este FE se calcula: **(altura subárbol derecho – altura subárbol izquierdo).** El árbol, no pierde su condición de AVL si el FE de cada nodo es: **-1, 0** ó **1**

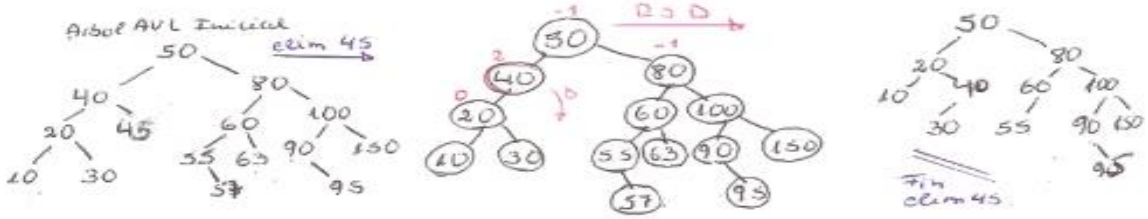
(Se elimina y agrega según propiedades de un **ABB**, si se produce desequilibrio, se resuelve con rotaciones cada vez que se realice una operación de insertar o eliminar un elemento)

3) Insertar en un árbol AVL, inicialmente vacío, la siguiente secuencia de enteros. Dibuje la traza con sus respectivas rotaciones: 3, 2, 18, 5, 20, 90, 77, 40, 34, 12
   En la traza, indicar el FE cuando se pierde la propiedad de AVL e indicar el tipo de rotación a aplicar:
   - **RSI:** Rotación simple a la izquierda
   - **RSD**: Rotación simple a la derecha
   - **RDlD:** Rotación doble a la derecha (entre medio se hace una rotación simple a la izq.)
   - **RDDI:** Rotación doble a la izquierda (entre medio se hace una rotación simple a la der.)

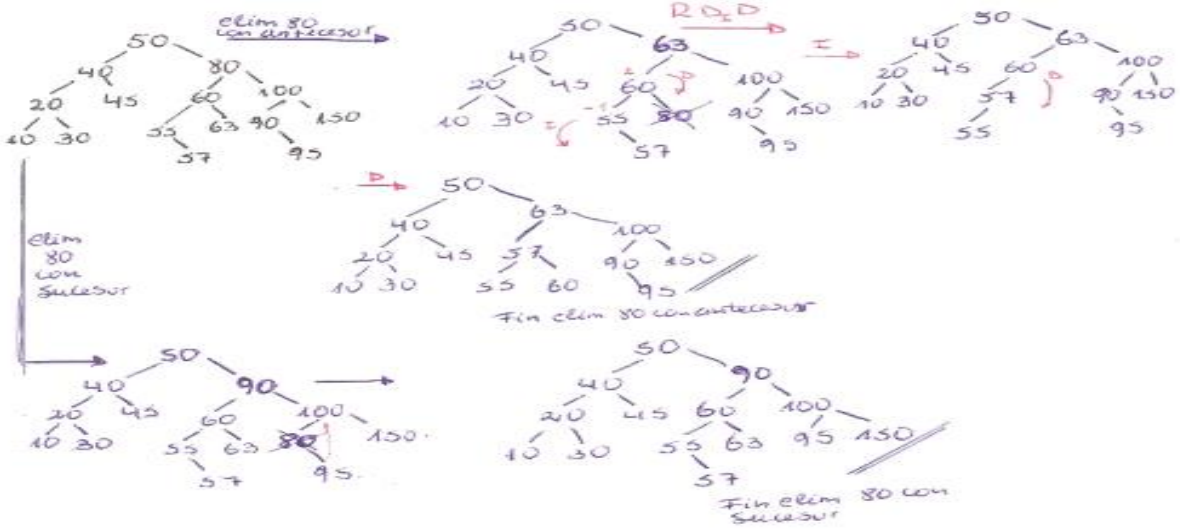


4) Para el siguiente árbol AVL: 50-80-100-150-40-45-20-10-30-90-95-60-55-57-63, haga las eliminaciones que se indican y utilice todas las posibilidades de eliminación según corresponda. El árbol debe seguir considerando su condición de AVL.
   a) Eliminar el 45
   b) Eliminar el 80

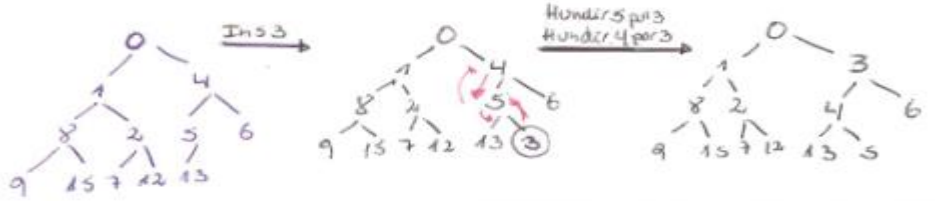
Eliminar 45:



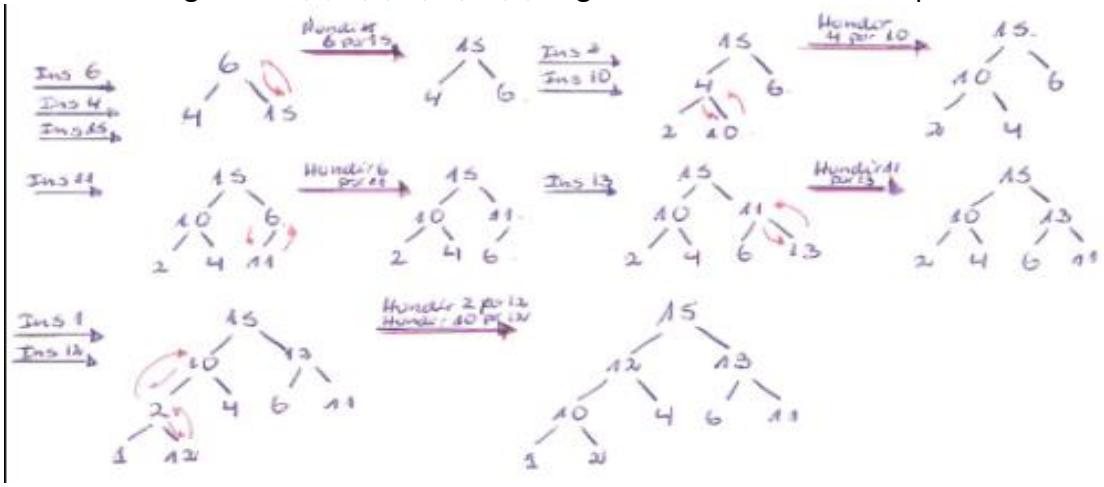
Eliminar 80:



5) Realice lo que se indica para cada caso de árboles binarios Heap:
   a) Para el Heap siguiente: 0,1,4,8,2,5,6,9,15,7,12,13, agregue el 3 y elimine un elemento.

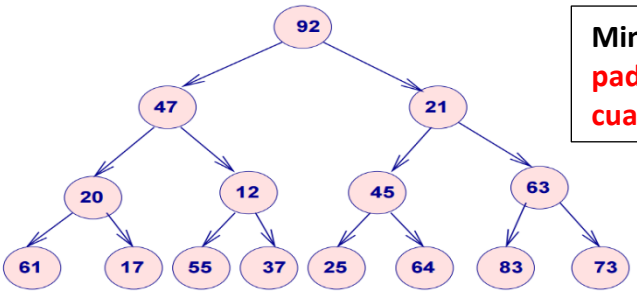


   b) Para la lista siguiente 6,4,15,2,10,11, 13,1,12 genere un árbol Max- Heap.



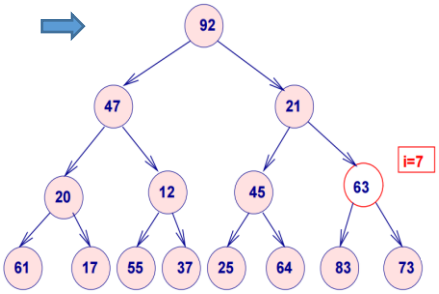
   c) Dado el siguiente árbol: 92-47, 92-21, 47-20, 47-12, 20-61, 20-17, 12-55, 12-37, 21-45, 21-63, 45-25,45-64, 63-83 y 63-73, realice la traza para que cumpla la propiedad de un Min-Heap. Cada traza debe contemplar las modificaciones del arreglo asociado al Heap.

Docente: Pamela Landero Sepúlveda



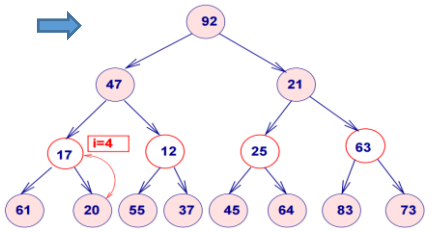
Min-Heap: El nodo padre es menor que cualquiera de sus hijos

Analizar nodos (padres) del penúltimo nivel hacia arriba, si existe un padre mayor a sus hijos **cambiar por el hijo que tenga el menor valor cumpliendo la propiedad de Min-Hep**. El arreglo se modifica considerando que para una posición **i** del arreglo, la posición hijo izquierdo es **2i**, la posición del hijo derecho es **2i + 1** y la del padre es **(int) i/2**



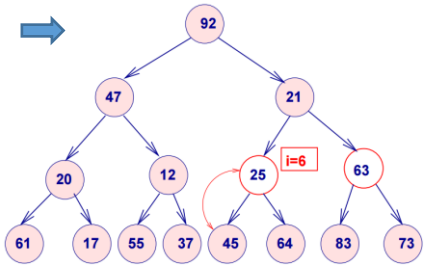
Está Ok para i=7

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
92	47	21	20	12	45	63	61	17	55	37	25	64	83	73



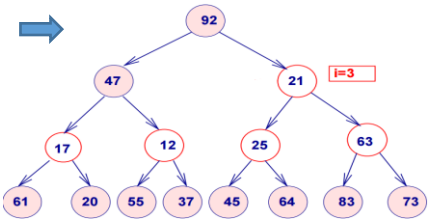
En i=4, se cambió el 17 por el 20. En pos **9** va el 20

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
92	47	21	17	12	25	63	61	20	55	37	45	64	83	73

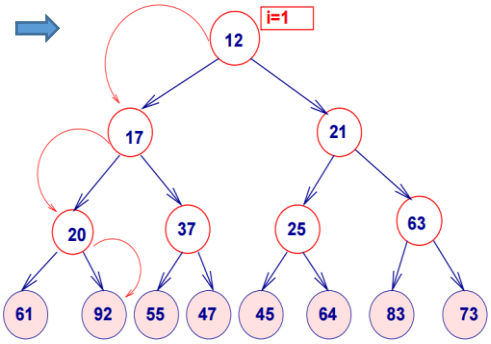


En i=3, se cambió el 45 por el 25. En pos **12** va el 45

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
92	47	21	20	12	25	63	61	17	55	37	45	64	83	73



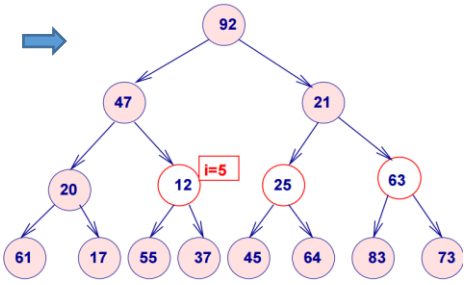
En i=3 está OK



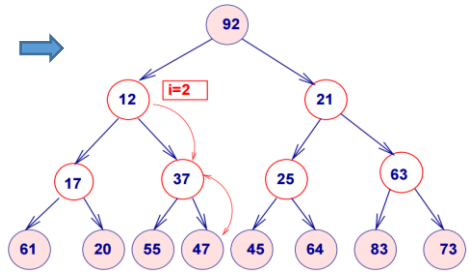
ÁRBOL FINAL

En i=1, se cambió el 92 por el 12. En pos **2** va el 92.
En i=2, se cambió el 92 por el 17. En pos **4** va el 92.
En i=4, se cambió el 92 por el 20. En pos **9** va el 92.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	92	21	17	37	25	63	61	20	55	47	45	64	83	73
12	17	21	92	37	25	63	61	20	55	47	45	64	83	73
12	17	21	20	37	25	63	61	92	55	47	45	64	83	73



En i =5 está OK



En i=2, se cambió el 47 por el 12. En pos **5** va el 47.
En i=5, se cambió el 47 por el 37. En pos **11** va el 47.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
92	12	21	17	47	25	63	61	20	55	37	45	64	83	73
92	12	21	17	37	25	63	61	20	55	47	45	64	83	73